

CROMEMCO

Cromix* Operating System

Addendum to the Instruction Manual

CROMEMCO, INC.
280 Bernardo Avenue
Mountain View, CA 94043

Part No. 023-4025

November 14, 1980

Copyright © 1980
CROMEMCO, INC.
All Rights Reserved

***trademark of Cromemco, Inc.**

Booting Up the Cromix Operating System

In order to bring up the Cromix Operating System, insert the Cromix Operating System disk in disk drive A and follow the procedure which is used for booting up CDOS (i.e., type four RETURN characters to cause the RDOS sign on to be displayed and then type B to boot the system - your procedure may vary depending on the disk controller board and switch settings being used).

The following prompt will be displayed:

```
Large floppy: (1,2,3,4)
Small floppy: (5,6,7,8)
Hard disk:    (9,10,11)
Enter root device number:
```

The first time the system is booted up, the proper response is 1 if drive A contains a large (8 inch) floppy disk or a 5 if drive A contains a small (5 inch) floppy disk. The hard disk may not be used until it has been initialized and the proper information has been copied to it.

The system will then come up with a discussion of how to copy the Cromix System disk. If the user is using a hard disk based system, these instructions should be followed indicating the destination drive as **hd0**, **hd1**, or **hd2** (drive E, F, or G). After this, the hard disk may be indicated as the root device when the system is booted up. The system may not be booted from the hard disk.

Cromemco strongly recommends that the Cromix System disk be copied for use and that the original disk be stored in a safe place.

When the user does not wish to read the instructions on copying disks each time the system is booted up, the first line of the file **/cmd/startup.cmd** may be deleted using the Screen Editor.

Important Note

The Cromix Operating System will not display more than one screen-full at a time without causing the terminal to BEEP to ask if the user is ready to continue. Type **CNTRL-Q** (hold the **CNTRL** key down and type **Q**) when you are ready to continue.

Copying the Cromix System Disk

This procedure will make a back up copy of the system disk. It will destroy all existing data on the disk to which the system is being copied. If it is necessary to update a system disk, refer to the next procedure, Updating the Cromix System Disk.

To make a copy of this disk, enter the command **NEWDISK** in response to the Cromix Operating System Prompt (#). The command **NEWDISK** must be followed by the name of the device on which the new disk is mounted. The names of the devices are:

fda = large floppy disk A
fdb = large floppy disk B
fdc = large floppy disk C
fdd = large floppy disk D

sfda = small floppy disk A
sfdb = small floppy disk B
sfdc = small floppy disk C
sfdd = small floppy disk D

hd0 = hard disk E
hd1 = hard disk F
hd2 = hard disk G

The first program which **NEWDISK** will execute is **INIT**. **INIT** will prompt the user to determine which disk is to be initialized. Respond to this prompt with the correct disk drive designator (A-G). Be sure to properly specify the correct disk drive as all data on the specified disk will be destroyed. All other questions may be answered by typing a **RETURN** as a response (thereby supplying the default answers).

The **NEWDISK** command will then execute several other programs. The message:

finished copying.

will be displayed when execution of the **NEWDISK** command is complete.

Updating a Cromix System Disk

If the Cromix Operating System has already been brought up and the user needs to update the system from a new floppy diskette, the copy tree utility may be used as follows:

1. Boot up on the new Cromix Operating System diskette.
2. Create a dummy directory and mount the existing Cromix Operating System disk or diskette in that directory:

```
# create /dummy  
# mount devname /dummy
```

In the above command, `devname` is a device name from the table in the Copying the Cromix System Disk section of this manual.

3. Use the copy tree utility program to copy all of the files from the new to the old disk:

```
# cptree -f / /dummy
```

The copy tree utility program will copy all of the files from the new disk to the existing system disk. The `-f` option will cause existing files to be replaced by new files with the same name (updated).

Using a Printer - Software Considerations

Output which is sent to the system printer will be displayed on the device to which the /dev/prt file is linked.

Dot Matrix Printer

The Cromix Operating System is shipped assuming a dot matrix printer (Cromemco Model 3703 or 3779) will be attached to the system. If this is not the case, or if the software has been altered, a dot matrix printer may be specified as follows:

```
# del /dev/prt
# maklink /dev/lpt1 /dev/prt
```

Fully Formed Character Printer

The following commands will specify that a fully formed character printer (Cromemco Model 3355 or 3355A) is the system printer:

```
# del /dev/prt
# maklink /dev/typ1 /dev/prt
```

Serial Printer

A serial printer driver is available under the Cromix Operating System. This driver utilizes an ETX/ACK protocol. The serial device is interfaced through a TU-ART with base addresses of 80H and 90H for devices A and B respectively. Port 91H is used to connect the serial device.

The serial device may be accessed through the device name /dev/ser1. The following commands can be used to specify that the serial device is the system printer:

```
# del /dev/prt
# maklink /dev/ser1 /dev/prt
```

In order to use the serial device, the baud rate must be established by using the mode utility program:

mode ser1 baud xxxx

where **xxxx** is one of the standard baud rates from 110 to 19,200.

Notes

The **/dev** directory may only be altered by a privileged user (such as **system**).

The file **\$LP** in Cromemco 32K Structured Basic and 16K Extended Basic is associated with the system printer. As such, all output to the **\$LP** file will go to the device linked to the **/dev/prt** file.

Drive/File Access From CDOS Programs

In order for CDOS programs to be able to access files which are on various drives, the CDOS Simulator converts disk specifiers to directory names. For example:

B:Filename becomes **/B/Filename**

If no disk specifier is used, or if the disk specifier A is used (as in **A:Filename**), the file is assumed to be in the current directory.

To take full advantage of this scheme, Cromemco recommends that a file structure be constructed as follows.

1. Create files B, C, D, etc. in the root directory. Each of these files corresponds to one of the disk drives in the system.
2. Mount each disk on the appropriate drive using the mount utility:

mount fdb /b

3. The files may then be read and written from CDOS programs. The CDOS Simulator, running under the Cromix Operating System, will automatically convert the CDOS drive specifiers to the appropriate directory names.
4. Each disk which is mounted must be unmounted before physically removing the disk from the system. To do this, use the unmount utility:

unmount fdb

Files created and used in this manner are Cromix Operating System format files and are not CDOS compatible. The only way to read and write CDOS format files from the Cromix Operating System is by using the Cdoscopy utility program.

Inode, Block, Track, & Cylinder Numbers

The following discussion and formulae pertain only to the Cromemco Hard Disk Drive.

Definitions

There are three **surfaces** on the hard disk to which data can be written. There are 350 concentric **cylinders**. Each cylinder includes three surfaces. The intersection of a surface and a cylinder is termed a **track**. A track is composed of 20 **sectors**. To summarize:

There are 3 surfaces per cylinder.
There are 20 sectors per track.
There are 60 sectors per cylinder.

Converting Inode Numbers to Block Numbers:

$$\frac{\text{inode} - 1}{4} + 20 = \text{block}$$

Converting Block Numbers to Cylinder, Surface, & Sector Numbers:

$$\frac{\text{block}}{\text{sectors per track}} = \text{quotient \& remainder}$$

The above division yields a quotient (whole number) and a remainder. The remainder is the **logical sector number**. Refer to the following table to convert this to the **physical sector number**. The quotient is used in the following division to obtain the cylinder and surface numbers:

$$\frac{\text{quotient}}{\text{surfaces per cylinder}} = \text{quotient2 \& remainder2}$$

The above division yields quotient2 which is the **cylinder number** and remainder2 which is the **surface number**.

Converting Logical and Physical Sector Numbers:

-----Sector-----
Logical Physical

1	1
2	4
3	7
4	10
5	13
6	16
7	19
8	2
9	5
10	8
11	11
12	14
13	17
14	20
15	3
16	6
17	9
18	12
19	15
20	18

-----Sector-----
Physical Logical

1	1
2	8
3	15
4	2
5	9
6	16
7	3
8	10
9	17
10	4
11	11
12	18
13	5
14	12
15	19
16	6
17	13
18	20
19	7
20	14

Converting Cylinder, Surface, and Sector Numbers to Block Numbers:

First the physical sector number must be converted to a logical sector number by using the table. Then the following formula will yield the block number:

$$((\text{cylinder} * \text{surfaces per cylinder}) + \text{surface}) * \text{sectors per track} + \text{logical sector} = \text{block}$$

Addendum to the Cromix Operating System

/etc/group

The file **/etc/group** must be present for the mail and passwd programs to operate properly with group parameters. This file has a format similar to the **/etc/passwd** file. The following fields appear on each line separated by colons. The line is terminated by a RETURN character.

1. group name
2. group password
3. group identification
4. user names of all users associated with the group, separated by commas.

Copy Utility

The Copy utility program has two options in addition to those mentioned in the Instruction Manual.

The **-t** option will cause a file to be copied only if:

1. the file does not exist in the destination directory, or
2. the source file has been modified more recently than the destination file. This comparison is done on a file by file basis.

The **-d** option will allow directory and device files to be copied. If this option is not specified, these types of files will not be copied.

Cptree Utility

The Cptree utility program has an option in addition to those mentioned in the Instruction Manual.

The **-t** option functions in the same manner as it does in the Copy utility described above.

In addition, a list of one or more file names may follow the source and destination directories:

cptree [-vft] source destination file-list

Only files whose names match at least one of the names in the file list will be copied. Ambiguous file names which are enclosed in quotation marks may be included in the file list.

L(list) Utility

The L utility program has two options in addition to those mentioned in the Instruction Manual.

The **-t** option sorts the list of files according to the last time each file was modified.

The **-r** option causes the specified sort to be performed in reverse order.

In addition, the meaning of the first column of numbers generated by the L utility has been changed in some cases. If the file which is listed is a regular file, the number associated with the file is still the number of bytes in the file. If the file is a directory, the number is the number of files in the directory. If the file is a device, the number is the device number.

Mode Utility - Function Keys

The Cromix Operating System will now handle Cromemco 3102 terminal function keys. The function keys may be enabled by calling the **Mode** utility as follows:

mode fn

This will cause the actual two byte sequence to be transmitted. For example, **control-B** and **p** are transmitted when function key 1 is depressed. When function key 2 is depressed, **control-B** and **q** are transmitted, and so forth.

Move Utility

The Move utility program has an option in addition to those mentioned in the Instruction Manual.

The **-t** option functions in the same manner as it does in the Copy utility described above.

Time and Date Utilities

The Time and Date utilities have two options which are not described in the Instruction Manual and they now function in a slightly different manner.

The time utility now includes the date. The **-s** option causes the user to be prompted for a new date and time. The **-e** option causes the date to be displayed in European fashion (dd/mm/yy).

Shell
command: **GOTO** or **GO**

purpose: This command causes control to be transferred within a command file.

summary: go label

arguments: line label

options: none

This command is used to transfer control within a command file. Control is transferred to the line specified by **label**. This can be used to repeatedly execute the same commands within a command file. When used in conjunction with the **if** and **shift** commands, the **goto** command can become part of a conditional loop with varying parameters.

A **line label** is any line within a command file which begins with a percent sign (%). Note that if a percent sign appears as a character other than the first character on a line, it indicates that the balance of the line is a **comment** and will be ignored by the Cromix Shell.

If the **goto** command is given with a non-existent line label, execution of the command file will be terminated.

Example:

```
%sample_label  
x  
y  
z % this is a comment  
goto sample_label
```

This sample command file will cause the commands **x**, **y**, and **z** to be executed repeatedly. The first line of the command file is a line label, as indicated by the leading percent sign.

Notice the use of the percent sign to indicate a comment on the fourth line of the file. The fifth (last) line of the file will cause control to be transferred to the specified label (**sample_label**).

Shell
 command: **IF**

purpose: This command conditionally executes another command.

summary: **if** -rewa filename command
 string-1 = string-2 command
 string-1 != string-2 command

arguments: access method and a file name

or

two strings separated by the equal
 (=) or not equal (!=) relational
 operator

and

a command line

options: none

The **if** command is used to conditionally execute another command. There are two basic forms of this command.

The first format of the **if** command listed above will execute **command** if the specified access method applies to the specified file.

The second and third forms of the **if** command will execute **command** if the specified relational condition is true. These formats of the **if** command **do not** require that the strings be enclosed in quotation marks but **do** require that the relational operator (= or !=) be located between two spaces.

The **if** command is most frequently used in conjunction with the **goto** command and is terminated by a semi-colon (;).

The following example demonstrates a method of checking for a null parameter:

```
if x#1 = x goto abcd
```

If **#1** is a null string, the left side of the equality will be equal to the string "x". If **#1** is not a null string, the left side of the equality will not be equal to the string "x". The right

side is always equal to the string "x". Thus, the equality will be true if #1 is a null string.

Shell
command: **REPEAT** or **REP**

purpose: This command repeats a command.

summary: rep count command

arguments: a count of the number of repetitions
command

options: none

This command is used to repeat a command a specified number of times.

Examples:

```
% repeat 5 echo "this line will be displayed five times"
this line will be displayed five times
this line will be displayed five times
this line will be displayed five times
this line will be displayed five times
this line will be displayed five times
%
```

Notice that the repeat command is terminated by a semicolon and that any command(s) following the semicolon will only be executed once. This means that the following command will display the date three times and then display the time once:

```
% repeat 3 date; time
```

Wednesday, November 12, 1980

Wednesday, November 12, 1980

Wednesday, November 12, 1980 Wednesday, November
12, 1980 18:54:04

Shell
command: **REWIND** or **REW**

purpose: This command restores the arguments
with which a command file was
called.

summary: rew

arguments: none

options: none

This command is used to restore the arguments with
which a command file was called. It nullifies the
effect of any **shift** commands given within the batch
file. After the execution of the rewind command,
#1 will represent the first argument with which the
command file was called, **#2** the second, etc.

Shell
command: **SHIFT**

purpose: This command shifts the arguments with which a command file is called.

summary: shift

arguments: none

options: none

This command is used to shift the arguments with which a command file is called. After the execution of the **shift** command, **#1** represents the second argument from the original command line, **#2** represents the third, etc. After another execution of the **shift** command, **#1** represents the third argument, etc.

The **rewind** command may be used to nullify the effects of the **shift** command.

Example:

```
%abc
screen #1
shift
if x#1 != x goto abc
```

If the above command file (named **abc.cmd**) is called as follows:

```
abc *.txt
```

the Shell will expand the ambiguous file name ***.txt** into a list of all of the files in the current directory which have the extension of **txt**. Then the command file **abc** will be called with this list as its arguments.

The first executable command within the command file is **screen**. Standard argument substitution will cause the **#1** to be replaced by the first argument from the command line.

After the execution of **screen**, the **shift** command will cause the second argument from the original command line to be represented by **#1**.

This argument from the command line will be substituted in place of **#1**. If this file exists,

the string `xfilename` will not be equal to the string `x` and control will be transferred to the line labelled `abc`. If this file does not exist, then a null string will be substituted for `#1`, the string `x` will be equal to the string `x`, and execution of the batch file will be terminated.

utility: **BACKUP**

purpose: This program will copy a directory and all subdirectories and files to a block device.

summary: backup [-tv] source-dir dest-dev file-list

arguments: source directory and destination device

optional list of one or more file names
Only files whose names match at least one of the names in the file list will be backed up. Ambiguous file names which are enclosed in quotation marks may be included in the file list.

options: -v verbose
This option will display pertinent information as files are backed up.

-t time
The -t option will cause a file to be backed up **only** if the source file has been modified since it was last backed up.

This program will copy the source directory along with all descendent directories and files to the destination device.

Disks for **backup** should be initialized for use with the Cromix Operating System. **Note that data existing on the disk in the destination device will be destroyed.**

If the destination device is a floppy disk and all of the data will not fit on one floppy disk, the **backup** program will prompt for additional disks.

The data which has been backed-up may be restored by the **restore** utility. A disk which has been written to by the **backup** utility may **only** be accessed by the **restore** utility.

utility: **RESTORE**

purpose: This program will restore data which has been saved by the backup utility.

summary: restore [-lv] source-dev [file-list]

arguments: source device and optional list of files to be restored

options: -l list only
 This option may only be used with the first (number one) disk in a set of backup disks. It will list the names of all of the files which are backed up on the set of disks.

-v verbose
 This option will display pertinent information as files are copied.

This program will recreate files which have been saved by the **backup** program.

The **restore** program always starts with the first disk (number 1) which was created by the **backup** program and prompts the user for additional disks from the set of backup disks as necessary.

If no file names are specified, the entire directory, including all descendent directories and files, will be restored to its original structure. If one or more file names are specified, the specified files will be restored **into the current directory**. If the optional file list is included, only files with names **exactly matching** those in the file list will be restored.

If the -l option is used, no restoration will take place.

